SYSTEM AND METHOD OF ACCESSING AND TRANSMITTING DIFFERENT DATA FRAMES IN A DIGITAL TRANSMISSION NETWORK

Field of the Invention

The present invention relates to a system and method of accessing and transmitting data frames, in particular to a system and method of accessing and transmitting different data frames in a digital transmission network.

Background of the Invention

Because Ethernet technology is featured with features low price and high expansibility, etc., it has evolved from a mainstream LAN technology to a primary data service access technology and is widely used in Metropolitan Area Network (MAN) by more and more telecom operators. Providing of Ethernet data services has becoming a trend for telecom operators. Ethernet data services can be classified into two types: Ethernet private line services and Virtual Local Area Network (VLAN) services.

For convenience, the phrases and abbreviations in the following description have the following meanings: MPLS - Multi-protocol Label Switching; GFP - General Frame Positioning; VLAN - virtual Local Area Network; VMAN - virtual Metropolitan Area Network; RPR - Resilient Packet Ring.

Currently, most of the telecom operators' data transmission networks are SDH/SONET networks. Therefore, it is a highlight for telecom operators and telecom equipment manufacturers to access and transfer Ethernet data frames effectively in a SDH/SONET network to meet the increasing demand for Ethernet data services. At present, several telecom equipment manufacturers have provided [[the]] devices to access and transfer Ethernet data frames in [[a]] an SDH/SONET network, and those devices may be classified into 3 types according to the implementation approach of functionality:

(1) Data mapping/demapping scheme;

(2) Bridge scheme;

(3) RPR scheme.

Fig.1 shows a block diagram of an apparatus ~~device~~ according to the data mapping/demapping scheme in the first prior art. The ~~device~~ apparatus ~~comprises~~ includes one or more subscriber network interfaces 20 (standard Ethernet interfaces), one or more inter-network interfaces 30 (synchronous digital transmission channels), one or more mapping/demapping devices 101, 102, ..., each of which corresponds to a unique subscriber network interface and a unique inter-network interface. Wherein, [[the]] data frames entering the ~~device~~ apparatus via the one or more subscriber network ~~interface~~ interfaces 20 and data frames output from the ~~device~~ apparatus comply with Ethernet data standard; data frames entering the apparatus ~~device vian~~ via the one or more inter-network interfaces ~~interface~~ 30 and data frames ~~frame~~ output from the apparatus ~~device~~ comply with synchronous digital transmission network standard.

Wherein the one or more mapping/demapping device ~~10 map~~ maps Ethernet data frames entering the apparatus ~~device~~ via the one or more subscriber network ~~interface~~ interfaces 20 to ~~become~~ synchronous digital data frames, and ~~outputs~~ output the mapped data frames [[vian]] via the one or more inter-network interfaces ~~interface~~ 30; further, the one or more mapping/demapping ~~device~~ devices ~~10 demaps the~~ demap synchronous digital data entering the apparatus ~~device vian~~ via the one or more inter-network ~~interface~~ interfaces 30 to Ethernet data frames, and ~~outputs the~~ output data frames via the one or more subscriber network interfaces 20. However, the functionality of the apparatus ~~device~~ is simple, thus it can only provide Ethernet private line services.

Fig.2A shows a block diagram of ~~the device~~ an apparatus utilizing the bridge scheme in the second prior art. The apparatus includes ~~device comprises~~ one or more subscriber network interfaces 20 (standard Ethernet interfaces), each of which corresponds to a unique bridge port. The ~~device~~ apparatus further ~~comprises~~ includes one or more inter-network interfaces 30 (synchronous digital transmission channels). The ~~device~~ apparatus further ~~comprises~~ includes a bridge device 400 (described in detail in IEEE802.1D and IEEE802.1Q), wherein the bridge device 400 ~~comprises~~ includes a plurality of bridge ports, each of which corresponds to a unique subscriber network

interface or a unique mapping/demapping device. Each mapping/demapping device corresponds to a unique bridge port and a unique inter-network interface. Wherein, [[the]] data frames entering the ~~device~~ apparatus via the one or more subscriber network interfaces ~~interface~~ 20 and [[the]] data frames output from the apparatus ~~device~~ comply with Ethernet data standard; data frames entering the apparatus ~~device~~ from the one or more inter-network ~~interface~~ interfaces 30 and data ~~frame~~ frames output from the apparatus ~~device~~ comply with the standard of synchronous digital transmission network.

The [[Data]] data frames entering the apparatus ~~device~~ via the one or more subscriber network ~~interface~~ interfaces 20 enter the bridge device 400 via [[the]] bridge ports [[port]] corresponding to the one or more subscriber network interfaces 20 interface[[;]]. [[the]] The bridge device 400 calculates [[the]] a bridge output port according to the address information in the data frames and sends the data frames to the corresponding mapping/demapping device 102 (the mapping/demapping device maps the data frames and then outputs them to the inter-network interface) via the output port, and vice versa.

In the bridge scheme, usually the operator is allowed to map partial or all subscriber network interfaces to mapping/demapping devices in a one to one way through configuration. In this case, the ~~device~~ apparatus employs both of above technical schemes, so it is called an enhanced bridge scheme. The functional model of an enhanced bridge device is shown in Fig.2B.

The disadvantage of the second prior art is:

(1) It is unable to provide integral VLAN service. If a plurality of subscribers are attached to the apparatus ~~device~~ via the one or more subscriber network interfaces and there are conflicts among address spaces of Ethernet data frames of those subscribers, the apparatus ~~device~~ is unable to isolate the conflicts effectively, thus it is unable to provide services correctly to those subscribers.

(2) A common bridge (non-enhanced bridge) is unable to provide Ethernet private line service.

(3) [[A]] One subscriber network interface can only support one service type (Ethernet private line service or VLAN service), which limits the access capability of

the apparatus device. In some cases, though the processing capacity of the device is still sufficient enough, new devices have to be added to improve access capacity because the subscriber network interfaces have been used up.

(4) [[An]] One inter-network interface can only support one service type (Ethernet private line service or VLAN service), which leads to low convergence capability of the apparatus device. In some cases, in a star topology network, though the processing capacity of the apparatus device is still sufficient enough, new devices have to be added to improve convergence capacity because the inter-network interfaces have been used up. For operators, it means not only new investment but also bandwidth waste.

Fig.3 shows a block diagram of the device an apparatus utilizing the RPR scheme in the third prior art. The apparatus device comprises-includes one or more subscriber network interfaces (standard Ethernet subscriber network interfaces), two inter-network interfaces (synchronous digital transmission channels), [[and]] [[a]] an RPR device 600 (described in IEEE802.17), two mapping/demapping devices, and a data processing device 500, which may be a data converging/deconverging converging/diverging device or a bridge device[[()]].

Wherein the data frames entering the apparatus device via the one or more subscriber network interface interfaces [[20]] are processed as follows:

Step 1: the data processing device 500 processes the data frames (the data frames are converged if the data processing device is a data converging/deconverging converging/diverging device; the data frames are switched if the data processing device is a bridge device);

Step 2: the data processing device 500 transfers the processed data frames to the RPR device 600;

Step 3: the RPR device 600 sends the data frames to the corresponding mapping/demapping device according to the address information in the data frames;

Step 4: the mapping/demapping device maps performs mapping operation for the data frames and sends them to outside of the apparatus device via the corresponding inter-network interface.

The data frames entering the apparatus device via the inter-network interface are

processed as follows:

Step 1: the mapping/demapping device performs demapping operation for the data frames and transfers the demapped data frames to the RPR device 600;

Step 2: the RPR device 600 processes the data frames and then sends them to the data processing device;

Step 3: the data processing device 500 processes the data frames (the data frames are ~~deconverged~~ diverged if the data processing device is a data ~~converging/deconverging~~ converging/diverging device; the data frames are switched if the data processing device is a bridge device);

Step 4: the data processing device 500 finds corresponding subscriber network interface according to the address information in the data frames and then outputs the data frames via the subscriber network interface.

The disadvantages of [[said]] the scheme are:

(1) It is unable to provide Ethernet private line service and VLAN service at the same time. If the data processing device is a bridge device, it doesn't support Ethernet private line service; if the data processing device is a data ~~converging/deconverging~~ converging/diverging device, it doesn't support VLAN service.

(2) It can only be used in a ring topology network.

Summary of the present invention

An object of the present invention is to provide a virtual bridge device in SDH/SONET network and the method of accessing and transmitting data frames in a SDH/SONET network thereof, in order to improve the converging ability.

According to one aspect of the present invention, it is provided a system accessing and transmitting different data frames in a digital transmission network for accessing and transmitting different data frames, [[said]] the system comprises:

At least a subscriber network interface, which is used to couple with the user's network; and/or at least an inter-network interface, which is used to couple with [[said]] the digital transmission network to transfer data; and

a data converting device, which is coupled with [[said]] the subscriber network interfaces and [[said]] the inter-network interfaces to convert data formats between [[said]] the subscriber network interfaces, data formats between [[said]] the inter-network interfaces, or data formats between [[said]] the inter-network interfaces and [[said]] the subscriber network interfaces;

Wherein [[said]] the data converting device comprises a virtual private device, an interface device and a processing device. [[Said]] The virtual private device exchanges data between [[said]] the subscriber network interfaces and [[said]] the inter-network interfaces via [[said]] the interface device. Wherein [[said]] the virtual private device comprises an inter-device interface, which couples with [[said]] the processing device and is used to input and output data frames; a virtual private processing unit, which couples with [[said]] the inter-device interface and is used to converge or deconverge diverge the data frames and detect control messages; a rule database, which couples with [[said]] the virtual private processing unit, [[said]] the rule database stores rules corresponding to various data, couples with [[said]] the virtual interface processing unit to process the data according to [[said]] the rules; a control interface unit, which couples with [[said]] the rule database and [[said]] the virtual private processing unit and is used to control [[said]] the virtual private processing unit and [[said]] the rule database.

Optionally, [[said]] the rule database stores convergence rule and deconvergence divergence rule.

Preferably, [[said]] the rule database stores relay rules.

Optionally, one data type corresponds to one rule, rule database comprises the following rules: input data frame type number, rule type (one of convergence, deconvergence divergence and relay rules), label number and output data frame type number.

Preferably, formats of the control messages and the logic processing [[said]] the data frames are stored in [[said]] the virtual private processing unit; and formats of rules are stored in [[said]] the rule database.

Optionally, [[said]] the control interface unit provides an external control interface, through which to inspect the operation of the virtual processing unit, and add, delete,

modify, and search operation can be performed to rules in [[said]] the rule databases.

Preferably, [[said]] the interface device is a virtual interface device, [[said]] the processing device is a data processing and dispatching device, [[said]] the inter-device interface connects with [[said]] the data processing and dispatching device or [[said]] the virtual interface device.

The present invention also provides a method of accessing and transmitting different data frames in a digital transmission network through [[said]] the virtual private device; [[said]] the system comprises virtual bridge devices, [[said]] the method comprises the following steps:

Data frames entering [[said]] the virtual private device via an inter-device interface;

determining whether the data frames are control messages;

If yes, sending the data frames to external control system via the control interface unit and end the process; If not, extracting input data type number information and search in the rule database according to [[said]] the input data type number;

Determining whether type number information is found;

If not, discard [[said]] the data frames and end the process;

If yes, process the data according to the rule type;

Modify the data frames, and send them through the inter-device interface and end the process.

Optionally, the step of processing the data according to the rule type comprises the following steps:

Determine the rule type,

If it is convergence rule, insert the label number defined in the rule in the special position of the data frames;

If it is deconvergence divergence rule, remove the label number in the special position of the data frames;

If it is relay rule, change the label number in the special position of the data frames into a label number defined in the rule.

Preferably, the step of modifying the data frames comprises the step of replacing

the data frame type number in the head position of the data frames with the output data frame type number defined in the rule.

The system and method according to the present invention have the following advantages:

(1) a plurality of convergence rules, ~~deconvergence~~ divergence rules and relay rules are stored in the virtual private device (relay rules are optional, i.e., relay rules are not used in some simple virtual private devices). One data type corresponds to one rule. Rules in the virtual private devices can be set up and deleted dynamically. Data frames of different users can be isolated, transmitted and shared by adding a label before sending, changing the label during transmission and removing the label at destination address.

(2) according to the system and method of the present invention, both VLAN service and Ethernet data private service can be achieved at a device interface (a subscriber network interface or an inter-network interface). And data frames of many subscribers can be accessed and converged at a device interface, thus the accessing and converging abilities of the device are improved greatly.

Brief Description of the Drawings

Fig.1 shows the block diagram of the data mapping/demapping scheme according to the first prior art;

Fig.2A shows the block diagram of the bridge scheme according to the second prior art;

Fig.2B shows the block diagram of the enhanced bridge scheme according to the second prior art;

Fig.3 shows the block diagram of RPR scheme according to the third prior art;

Fig.4 shows the schematic diagram of a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.4 (F) shows the data processing flowchart of a preferred embodiment of the

system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network;

Fig.4A shows the schematic block diagram of virtual interface device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in an SDH/SONET network according to the present invention;

Fig. 4A (F1) shows the flowchart of the virtual interface device processing the data entering from the interface of processing device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.4A (F2) shows the flowchart of the virtual interface device processing the data entering from the inter-device interface in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.5 shows the schematic block diagram of the virtual private device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.5 (F) shows the flowchart of the virtual private device processing the data entering from the inter-device interface in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.6 shows the schematic block diagram of the virtual bridge device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.6 (F1) shows the flowchart of the virtual bridge device processing the data entering from the inter-device interface in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.7 shows the schematic block diagram of the data processing and dispatching device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Fig.7 (F) shows the flowchart of the data processing and dispatching device processing the entered data in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention;

Detailed Description of the Embodiments

To make the system and method of the present invention be better understood, the basic function of the device is describe first. Usually, any device has the following 3 functions:

(1) Input function, i.e., receive external information;

(2) Processing function, i.e., process [[said]] the external information;

(3) Output function, i.e., output the processed information.

For a device accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network:

(1) Input function: receive external information via the subscriber network interfaces and the inter-network interfaces;

(2) Output function: output the processed information via the subscriber network interface and the inter-network interface;

(3) Processing function: for a device accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network, different processing ability abilities as well as service ability and service efficiency depend on different technical schemes.   .

In addition, the phrase "standard Ethernet interface" used in the present invention means the following:

IEEE802.3 defines the LAN interface in detail[[;]]_ Such an interface is regarded as a standard Ethernet interface in the present invention.

The present invention will be described hereunder with reference to the drawings. For concision, components and units described in prior art will not be described in detail

hereunder. And components and units described above will not be described in detail hereunder.

Fig.4 shows a system that accesses and transmits different data frames in a digital transmission network. The system comprises a plurality of subscriber network interfaces designed to couple with the subscribers' networks, a plurality of inter-network interfaces designed to couple with [[said]] the digital transmission network to transmit data frame, a plurality of mapping/demapping devices 10, a virtual interface device 80 coupled with [[said]] the plurality of subscriber network interfaces and also coupled with the plurality of [[said]] inter-network interfaces via the plurality of [[said]] mapping/demapping devices 10, a data processing and dispatching device 90 coupled with [[said]] the virtual interface device 80, a plurality of virtual private devices device 120 and a plurality of virtual bridge [[device]] devices 100 and a PRP device 110 coupled with [[said]] the data processing and dispatching device. Wherein the plurality of [[said]] mapping/demapping devices 10, [[said]] the virtual interface device 80, and [[said]] the data processing and dispatching device 90 constitute the data converting device of the present invention. The input to [[said]] the system comprises includes: (1) data frames entering the system via the plurality of subscriber network interfaces; (2) data frames entering the system via the plurality of inter-network interfaces. The data frames output from [[said]] the system comprises includes: (1) data frames from the plurality of subscriber network interfaces; (2) data frames from the plurality of inter-network interfaces.

Fig.5 shows the schematic block diagram of a virtual private device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention. The system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network comprises includes a plurality of subscriber network interfaces designed to couple with the subscribers' networks, a plurality of inter-network interfaces designed to couple with [[said]] the digital transmission network to transmit data frames, a plurality of mapping/demapping devices 10, a virtual interface device 80 coupled with the plurality of [[said]] subscriber network interfaces and also coupled with the plurality of [[said]] inter-network

interfaces via the plurality of [[said]] mapping/demapping devices 10, a data processing and dispatching device 90 coupled with [[said]] the virtual interface device 80, a plurality of virtual private devices device 120 and a plurality of virtual bridge [[device]] devices 100 and a PRP device 110 coupled with [[said]] the data processing and dispatching device. In the virtual private device, a virtual private processing unit 8001 is coupled to [[the]] an inter-device interface to process [[the]] data frames from the inter-device interface. The virtual private processing unit 8001 is also coupled to [[the]] a rule database and [[the]] a control interface unit. The control interface unit exchanges data frames with outside world via [[the]] a control interface.

The present invention enhances the converging ability of the system through utilizing the virtual private device. The virtual private device stores converging rules, deconverging diverging rules, and relay rules (the relay rules are optional and unnecessary in some simple virtual private devices). The mapping relationship between data types and the rules is 1:1. The rules in the virtual private device can be set up and deleted dynamically. Data frames of different subscribers can be isolated, transmitted and shared by adding a label before sending, changing the label during transmission and removing the label at destination address. A virtual private device comprises a virtual private processing unit and a rule database in it.

The virtual private processing unit has two major functions:

(a) Detecting control messages and transferring the control messages to [[the]] a control system of the virtual private device via the control interface unit.

(b) Performing convergence, deconvergence divergence, or relay operation for data frames except for the control messages.

The virtual private processing unit is the processing center of the virtual private device; the following items are stored in the virtual private processing unit as firmware:

(a) Format of control messages;

(b) Processing steps and logic for data frames;

(c) Format of rules in the rule database;

As [[the]] a control center, the rule database controls the processing action of the virtual private processing unit. The rules in the rule database may be updated

dynamically. The rule database may store a plurality of rules, each of which comprises the following information: input data frame type number, rule type (convergence/~~deconvergence~~divergence/relay), label number, and output data frame type number.

Wherein [[said]] the virtual private device is connected to the [[said]] data processing and dispatching device via the inter-device interface; [[said]] the virtual private device is connected to the control system of the device via the control interface.

There are different technical schemes to implement the virtual private device. However, the ~~data frames~~ processing steps and logic for data frames in the virtual private processing unit [[is]] are identical under these schemes. The main differences between these schemes are:

(a) Format of rules in the rule database, for example, the length and position of a label in the rule are different;

(b) Format of control messages processed in the virtual private processing unit.

In view of expandability and compatibility, it is recommended to implement the virtual private device with MPLS, GFP, VMAN, or Nested VLAN technique. The device manufacturers may also employ self-defined label formats (or self-defined data frame packets) to implement [[said]] the virtual private device. The system may support a plurality of virtual private devices implemented with different technical schemes.

Fig.5 (F) shows the flowchart of the virtual private processing; for data frames entering the virtual private device via the inter-device interface, the processing steps of the virtual private processing unit are as follows:

Step 1: determining whether the data frames are control messages; if [[yes]] not, going to step 3;

Step 2: transferring the data frames to an external control system via the control interface unit, and then going to step 12;

Step 3: extracting [[the]] an input data type number ~~information~~ from the head position of the data frames;

Step 4: searching in the rule database with the index of the input data type number ~~information~~;

Step 5: determining [[the]] a searching result; if it is blank, discarding [[said]] the data frames and going to step 12;

Step 6: determining [[the]] a rule type; if it is a convergence rule, going to step 7; if it is a ~~deconvergence~~ divergence rule, going to step 8; if it is a relay rule, going to step 9;

Step 7: modifying the data frames, i.e., inserting a label number defined in the rule at a special position [[in]] of the data frames, and then going to step 10;

Step 8: modifying the data [[frame]] frames, i.e., removing [[the]] a label number at [[the]] a special position [[in]] of the data frames, and then going to step 10;

Step 9: modifying the data frames, i.e., replacing [[the]] a label number at [[the]] a special position [[in]] of the data frames with a label number defined in the rule;

Step 10: modifying the data frames, i.e., replacing the data type number at the head position of the data frames with [[the]] an output data type number defined in the rule;

Step 11: transferring the data frames to the data frame processing and dispatching device via the inter-device interface;

Step 12: ending.

Fig.4 [[(F1)]] (F) shows the processing steps for data frames entering the system via the subscriber network interfaces.

First, in step 1, the virtual interface device performs matching action to the data frames according to classifying rules;

In step 2, the virtual interface device modifies the data frames according to classifying rules, i.e., inserts a data type number in the data frames;

In step 3, the virtual interface device transfers the modified data frames to the data processing and dispatching device;

In step 4, the data processing and dispatching device finds a corresponding processing device according to the data type number in the data frames;

In step 5, the data processing and dispatching device transfers the data frames to the corresponding processing device; if it is the virtual private device to process the data frames, the data frames are transferred to the virtual private device; if it is the virtual interface device to process the data frames, goes to step 8;

In step 6, [[said]] the corresponding processing device processes the data frames, and modifies the data type number at the end of the processing, and then transfers the modified data frames to the data processing and dispatching device;

In step 7, the procedure goes to step 4;

In step 8, the virtual interface device finds a corresponding device interface according to the data type number in the data frames;

In step 9, the virtual interface device modifies the data frames, i.e., deletes the data type number from the data frames;

In step 10, the virtual interface device outputs the modified data frames via the device interface (if the device interface is [[a]] an inter-network interface, mapping operation should be performed through the mapping/demapping device before output).

The processing steps for data frames entering the device via the network-network interface are as follows:

In step 1, the mapping/demapping device performs demapping operation to the data frames;

In step 2, the remaining processing steps are identical to those for data frames entering the system via the subscriber network interfaces.

When implementing functions of the system, the device manufacturers may partially or completely employ above method. For devices with simple functions, the processing is relatively simple. The device manufacturers may even treat the processing as firmware through fixing connections among the devices to omit the data processing and dispatching device.

Fig. 4A shows the schematic block diagram of virtual interface device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in an SDH/SONET network according to the present invention. [[A]] The virtual interface device 80 is used to enhance access ability. Via the virtual interface device 80, [[a]] one device interface (subscriber network interface 20 or inter-network interface 30) may be expanded to a plurality of virtual interfaces, each of which supports specific subscribers and services. For data frames entering the system via the device interfaces (subscriber network interface 20 and inter-network interface 30), [[the]] a virtual interface

processing unit 800 of the virtual interface device will classify them according to different services required by the subscribers and choose corresponding processing for them. Different processing corresponds to different rules in the rule database 850. [[The]] A control interface unit controls the virtual interface processing unit 800 to classify the data frames according to [[the]] orders from [[the]] a control interface and searches for corresponding rules stored in the rule database 850 to process the data frames. Data frames output from the device are transferred to a corresponding device interface after classified in the virtual interface device. Because the virtual interface device stores N rules, the mapping relationship between the device interface and those rules is 1:N. The rules in the virtual interface device may be set up or deleted dynamically. Therefore, it is easy to modify the rules to enhance the adaptability of the system according to the subscribers' requirements and the updating of the system. The device also comprises a software loader (not shown) to load different software.

As [[the]] a processing center, the virtual interface processing unit 800 is responsible for processing the data frames; the processing steps and processing logic in the virtual interface processing unit are firmware and can't be modified during operation of the system. The rule database is [[the]] a control center and is responsible for providing relevant processing and control parameters when the virtual interface processing unit processes the data frames. Different parameters lead to corresponding processing behaviors. During operation of the system, the rules in the rule database may be updated. The control interface unit provides an external control interface. Via the control interface, the control system of the device may monitor the virtual interface processing unit and performs adding, deleting, editing, and querying operations to rules in the rule database. The rule database may store a plurality of rules, each of which contains five parts: device interface number, data frame type number, data frame address offset, data frame type value, and data frame comparison mask.

Wherein the virtual interface device is connected to the subscriber network interfaces or the inter-network interfaces via the device interfaces. The mapping relationship between the subscriber network interfaces (or inter-network interfaces) and the device interfaces is 1:1. The virtual interface device is connected to the data

processing and dispatching device via the inter-device interface. The virtual interface device is connected to the control system of the device via the control interface.

Fig.4A (F1) shows the steps of the virtual interface processing unit processing data frames entering the virtual interface device via the device interface:

Step 1: searching for the first rule in the rule database corresponding to the device interface with the index of the device interface number;

Step 2: determining the searching result; if it is blank, discarding [[said]] the data frames and going to step 10;

Step 3: reading information at address offset of the data frame according to the data frame address offset in the rule;

Step 4: performing "AND" operation between [[said]] the read information and the data frame comparison mask in the rule;

Step 5: comparing the result of step 4 with the data frame type value in the rule; if they are equal, going to step 8;

Step 6: searching for the next rule corresponding to [[said]] the device interface in the rule database;

Step 7: going to step 2;

Step 8: modifying the data frames frame, i.e., inserting a data type number information in the rule in the head position of the data frames;

Step 9: transferring the data frames to the data frame processing and dispatching device via the inter-device interface;

Step 10: ending.

Fig.4A (F2) shows the steps of the virtual interface processing unit processing data frames entering the virtual interface device via the inter-device interface. The processing steps are as follows:

Step 1: extracting the data frame type number information from a head position of the data frames;

Step 2: searching in the rule database with the index of data frame type number;

Step 3: determining the searching result; if it is blank, discarding [[said]] the data frames and going to step 6;

Step 4: modifying the data frames, i.e., deleting the data frame type number information-from the head position of the data frames;

Step 5: sending the data frames to a corresponding device interface according to the device interface number in the rule;

Step 6: ending.

Fig.7 shows the schematic block diagram of the data processing and dispatching device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention. In [[The]] the data processing and dispatching device, a data processing and dispatching unit is coupled with a plurality of inter-device interfaces to process [[the]] data frames from the inter-device interfaces. The data processing and dispatching unit device-is also coupled with [[the]] a processing steps database and [[the]] a control interface unit. The control interface unit exchanges data with outside world via [[the]] a control interface.

The present invention utilizes the data processing and dispatching device to implement individualized services to improve equipment serviceability. The data processing and dispatching device stores a plurality of processing flows. The mapping relationship between the processing flows and [[the]] data frame types is 1:1. The data processing and dispatching device finds a corresponding processing flow according to a data frame type, and informs other devices to process the data frames according to the processing flow. The processing flows in the data [[frame]] processing and dispatching device may be set up, edited, or deleted dynamically. During operation of the system, the operator may quickly provide individualized services to maximize the efficacy of the system through adding, editing, and deleting the processing flows in the data processing and dispatching device.

The data processing and dispatching device is [[the]] a processing center of the system, and the processing flows are embedded in it as firmware. There are a plurality of inter-device interfaces in the data processing and dispatching device, each of which maps to a unique external device. The mapping relationship between the inter-device interfaces and the external devices are fixed.

The processing flow database is [[the]] a control center of the system, and the

items in the database may be updated dynamically. Each processing flow in the processing flow database contains the following information:

(a) Data frame type number;

(b) Inter-device interface number

The data processing and dispatching device is connected to other devices via the inter-device interfaces, each of the inter-device interfaces maps to [[a]] one virtual bridge device, [[a]] one virtual private device, [[a]] one RPR device, or [[a]] one virtual interface device. The data processing and dispatching device is connected to the control system thereof of the device via the control interface.

Fig.6 shows the schematic block diagram of a virtual bridge device in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention. The present invention overcomes the issue of limited Ethernet data frame address space through [[a]] the virtual bridge [[device]] devices. [[A]] One virtual bridge device may provide a plurality of virtual bridges. Each virtual bridge possesses all features and properties of the bridge device. However, it is different to the bridge device that the virtual bridge may be set up and deleted dynamically. During operation of the system, the operator may set up or delete a plurality of virtual bridges dynamically. because Because each virtual bridge has an independent address space, the operator may provide VLAN services to subscribers with different virtual bridges if [[that]] conflicts [[exists]] exist among address spaces.

The operator may configure a virtual bridge just like a physical bridge. The virtual bridge further expands the features of the physical bridge to support VMAN-based switching operation. [[A]] Each of the virtual bridge [[device]] devices comprises a virtual bridge processing unit 802, a multicasting database 856, a control interface unit, a forwarding database 852, and a virtual bridge database 854.

The virtual bridge processing unit has 3 major functions:

(a) Detecting control messages and transferring the control messages to [[the]] a control system of the device via the control interface unit.

(b) Learning addresses, and storing the knowledge acquired into the forwarding

database;

(c) Performing switching for data frames except for control messages and modifying data type information in the data frames.

The virtual bridge processing unit is [[the]] a processing center of the virtual bridge device, and there are five items embedded in it:

(a) Format of control messages;

(b) Processing steps and logic for data frames;

(c) Format of forwarding items in the forwarding database;

(d) Format of multicasting items in the multicasting database;

(e) Format of items in the virtual bridge database

The virtual bridge database, multicasting database, and forwarding database control [[the]] processing behaviors of the virtual bridge processing unit, and the items in them may be updated during operation of the system. The data formats of items in the multicasting database items and forwarding database are identical, and each item contains the following information:

(a) virtual Virtual bridge number;

(b) Input port of virtual bridge;

(c) Input Destination address [[input]];

(d) Input VLAN serial number [[input]];

(e) Input VMAN serial number [[input]];

(f) Output port of virtual bridge;

In the multicasting database, the database code is the combination of all fields; in the forwarding database, the database code is the combination of all fields except for output port of virtual bridge.

Each item in the virtual bridge database contains the following information:

(a) Input Type number [[input]];

(b) [[virtual]] Virtual bridge number;

(c) Port number;

(d) Output Type number [[output]];

Via the control interface, an external control system may implement the following

functions:

(a) Performing adding, deleting, editing, and querying operations to items in the database;

(b) Monitoring the working state of the virtual bridge processing unit.

The virtual bridge device is connected to the data processing and dispatching device via the inter-device interface. The virtual bridge device is connected to the control system of the device via the control interface.

Fig.6 (F1) shows the flowchart of the virtual bridge device processing the data entering the inter-device interface in a preferred embodiment of the system accessing and transmitting Ethernet data frames in a SDH/SONET network according to the present invention. For data frames entering the device via the inter-device interface, the virtual bridge processing unit performs the following processing steps:

Step 1: determining whether the data frames are control messages; if yes, going to step 3;

Step 2: transferring the data frames to an external control system via the control interface unit, and then going to step 17;

Step 3: extracting an input data type number, a destination address, a source address, a VLAN number, and a VMAN number (optional) from a fixed position in the data frames;

Step 4: searching in the virtual bridge database with the index of the data type number extracted;

Step 5: determining [[the]] a searching result; if it is blank, discarding [[said]] the data frames and going to step 17;

Step 6: obtaining the virtual bridge number and port number information from the searching result;

Step 7: learning the source address, and then updating the forwarding database according to [[the]] a learning result;

Step 8: determining whether the destination address is a multicasting address; if yes, executing a multicasting sub-flow, and then going to step 17;

Step 9: determining whether the destination address is a broadcast address, if yes,

~~executes~~ executing a broadcasting sub-flow, and then going to step 17;

Step 10: searching in the forwarding database with the index of virtual bridge number, input port, and destination address, VLAN number, and VMAN number (optional);

Step 11: determining the searching result; if it is blank, executing a broadcasting sub-flow, and then going to step 17;

Step 12: extracting output port number from the searching result;

Step 13: searching in the virtual bridge database with the index of the virtual bridge number and the output port number;

Step 14: determining [[the]] a searching result; if it is blank, discarding [[said]] the data frames and going to step 17;

Step 15: extracting an output type number from the searching result and modifying the data ~~frame~~ frames, i.e., replacing the type number in the data frames with the output data type number;

Step 16: outputing the modified data frames via the inter-device interface;

Step 17: ending.

Fig.7 (F) shows the flowchart of the data processing and dispatching device processing the data frames entering the [[device]] system in a preferred embodiment of the system accessing and transmitting Ethernet data frames in [[a]] an SDH/SONET network according to the present invention. The processing steps are as follows:

Step 1: extracting [[the]] a type number from the data frames;

Step 2: searching in the processing flow database with the index of the extracted type number;

Step 3: determining [[the]] a searching result; if it is blank, discarding [[said]] the data frames and going to step 6;

Step 4: extracting an inter-device interface number from the searching result;

Step 5: outputting the data frames via [[the]] an inter-device interface corresponding to the inter-device interface number;

Step 6: ending.

Application of the system:

The operator may utilize the system shown in Fig.4 to provide users Ethernet services, which correspond to combinations of processing flows. The operator may define and choose required processing flow combinations flexibly according to network topology, the users' service demands and bandwidth resource. Processing flow combinations common used are as follows:

Processing flow combination 1:

(1)The virtual interface device classifies the data frames;

(2)The virtual interface device outputs the data frames to corresponding device interfaces.

Processing flow combination 2:

(1) The virtual interface device classifies the data frames;

(2) The virtual bridge ~~Device~~ device switches the data frames;

(3) The virtual interface device outputs the data frames to corresponding device interfaces.

Processing flow combination 3:

(1) The virtual interface device classifies the data frames;

(2) The virtual private device processes the data frames (relay, converge, or ~~deconverge~~ diverge);

(3) The virtual interface device outputs the data frames to corresponding device interfaces.

Processing flow combination 4:

(1) The virtual interface device classifies the data frames;

(2) The RPR device processes the data frames (terminates, relays, or starts data frame transmission);

(3) The virtual interface device outputs the data frames to corresponding device interfaces.

When implementing functions of the system, the device manufacturers may partially or completely employ above method. For devices with simple functions, partial components can be omitted and the processing is relatively simple. The device manufacturers may even treat the processing as firmware through fixing connections

among the devices to omit the data processing and dispatching device. For instance, for a device that only provides Ethernet private service, it may only comprises the following devices:

(1) One or more subscriber network interfaces;

(2) One or more inter-network interfaces and mapping/demapping devices;

(3) A virtual interface device;

(4) A virtual private device.

[[Said]] The device supports the following processing flow combinations:

(1) Processing flow combination 1;

(2) Processing flow combination 3.

With the system and method of the present invention, integral VLAN services can be provided for subscribers. The system and method in the present invention overcome the restriction on Ethernet data frame address space at subscriber network interfaces. When a plurality of Ethernet data frames is sent to such a network device, there is no restriction on address space of Ethernet data frames.

With the system and method of the present invention, a device interface (subscriber network interface or inter-network interface) supports both VLAN service and Ethernet private service, and a plurality of Ethernet data frames from many subscribers can be accessed and converged at a single device interface, thus the accessing and converging capability of the device is enhanced greatly.

With the system and method of the present invention, the operator may maximize processing capacity of its equipments and utilization ratio of bandwidth resource through flexible adjustment to internal processing flows according to network topology, the subscribers' service demands, and bandwidth resource.

With the system and method of the present invention, individualized services can be quickly provided for subscribers. The operator may provide new services to subscribers through creating new flow combinations, without upgrading or adding any new equipment.

According to the present invention, data frames entering the device via device interfaces are processed after classified; different data types correspond to different

processing flows, which are implemented by the virtual interface devices.

Existing technical schemes and devices have to be implemented on a presumption: data frames from [[a]] one device interface belong to the same type, and different types of data frames enter the device via different device interfaces. Therefore, in the prior technical scheme and system, data frames entering the device via the same device interface are processed through the same processing flow, thus the mapping relationship between processing flows and device interfaces is 1: n. As a result, if there are many data types sometimes, though the processing capacity of the device is still sufficient enough, some services can't be supported due to lack of free device interfaces.

Whereas with the system and method of the present invention, data frames from [[a]] one device interface may contain various types, and processing flows for different types of data frames are different; the mapping relationship between processing flows and device interfaces is m: n.

In addition, the present invention supports free combination of processing flows; the combination of data processing and dispatching device and its processing flows is an important part of the present invention.

Existing technical schemes and devices have no data processing and dispatching device and only support several fixed processing flows. As the result, they are lack of flexibility and resiliency. However, the requirements of operators for equipment are changing actually.

A system lacking of flexibility and resiliency is unable to adapt to changing environment. With the present invention, the operator can update the processing flows dynamically, and subscribers can choose appropriate flow combinations or create new flow combinations as required. There are 22 or more processing flow combinations available.

In existing technical schemes and devices, the amount of bridge devices is certain; so conflicts among data frame address spaces of different subscribers can be avoided only through restricting the data frame address spaces of subscribers. Whereas the virtual bridge devices in the present invention support dynamical expansion (i.e., add new virtual bridges), and the address space of each virtual bridge is independent;

different virtual bridges correspond to different subscribers; hence the present invention has no restriction to data frame address spaces of subscribers.

In addition, Data frames of different subscribers can be isolated, transmitted and shared by adding a label before sending, changing the label during transmission and removing the label at destination address, thus the virtual private device of the present invention is achieved.

Though the present invention is described with reference to above embodiments, it is understood that any skilled in this field can easily make any change and modification without escaping the spirit of the present invention. So any change and modification should be in the scope of the present invention.